REMARKS

Claims 1, 4-9 and 11-19 are pending in the application. Claims 1 and 5 have been amended herein, and claim 10 has been newly canceled. Claims 15-19 have been newly added. Favorable reconsideration of the application is respectfully requested.

Applicants initially wish to thank the Examiner for the continued careful examination.

I. REJECTION OF CLAIMS 1 AND 12 UNDER 35 USC §103(a)

Claims 1 and 12 now stand rejected under 35 USC §103(a) based on *Westheimer et al.* in view of *Buerkle et al.* Applicants respectfully request withdrawal of the rejection for at least the following reasons.

Applicants have amended claim 1 to recite more particularly the function of the CPU in relation to the RAM and ROM in accordance with the present invention. More particularly, applicants have amended claim 1 to recite, *inter alia*, that the CPU is "configured to *judge whether intermediate code obtained from the RAM is the intermediate code or the encrypted intermediate code*, independent *of where the intermediate code is stored in the RAM*. Amended claim 5 and new claim 17 recite similar features. Neither *Westheimer et al.* nor *Buerkle et al.*, whether taken alone or in combination, teach or suggest such features as explained in more detail below.

i. Claim Amendments

Figs. 6 and 8 of the present application, reproduced below, illustrate the manner in which the LSI/optical disc apparatus of the present invention judge whether intermediate code obtained from memory is the intermediate code or the encrypted

intermediate code independent of where the intermediate code is stored in memory. Such feature is significant in that the present invention allows a user to create and/or modify intermediate code within the LSI/optical disc with relative ease. Moreover, the encrypted intermediate code may represent vendor proprietary command control strings that are not intended to be modifiable by a user. For reasons described more fully below, this is significantly different from other devices which include non-encrypted and encrypted code in memory.

FIG. 6

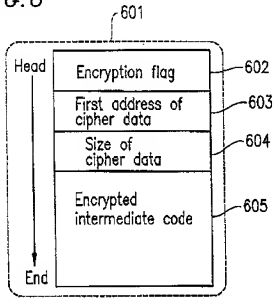

FIG. 8

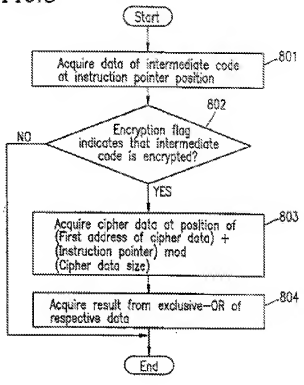

Fig. 6 of the present application exemplifies how intermediate code 601 stored in memory is identified as intermediate code or encrypted intermediate code based on the presence of an encryption flag 602 within the header. As is described in the present specification, for example, if the intermediate code is encrypted, data 0x01 is recorded as the encryption flag 602. If the intermediate code is not encrypted, data 0x00 is recorded as the encryption flag 602. (See, e.g., Spec., p. 19, Ins. 8-18).

Fig. 8 of the present application exemplifies how the intermediate code stored in memory is obtained from memory, for example at an instruction pointer position within the memory. In step 802, it is determined whether or not the intermediate code has been encrypted by referring to the value of the encryption flag 602. If the intermediate code is judged to be the encrypted intermediate code based on the flag 602, the process proceeds to steps 803, 804 in order to decrypt the intermediate code. If the intermediate code is judged not to be encrypted in step 802 based on the flag 602, the decryption steps 803, 804 are skipped.

Thus, according to the present invention whether the intermediate code is judged as intermediate code or encrypted intermediate code is independent of where the intermediate code is stored in memory, as recited in claims 1, 5 and 17. For example, whether the intermediate code is judged as intermediate code or encrypted intermediate code is based on other information such as the header information (e.g., flag 602), as recited in new claims 15-16 and 18-19.

ii. Westheimer et al.

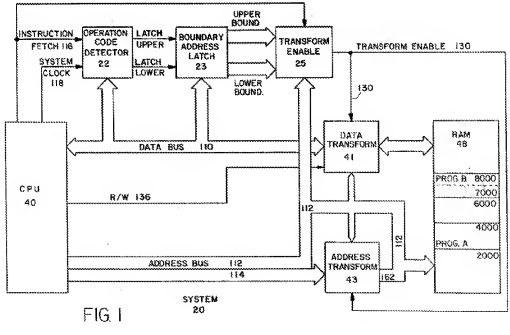

FIG. 1

Westheimer et al. describes a computer software protection system as illustrated in Fig. 1 of *Westheimer et al.* (reproduced above). The Examiner refers to Fig. 1 of *Westheimer et al.* as teaching a RAM 48 for storing an intermediate code representing a command control string to be executed by a control section and an encrypted intermediate code representing another command control string to be executed by the control section after first being decrypted.

Applicants acknowledge that *Westheimer et al.* teaches that the RAM 48 includes a non-encoded program A and a protected (encoded) program B. However, applicants respectfully note that the programs A and B in RAM 48 do not constitute intermediate code as recited in claims 1 and 12. Rather, the programs A and B stored in the RAM 48 represent operation code (i.e., machine code instructions). (See, e.g., Col. 5, lns. 50-53). As understood by those having ordinary skill in the art, intermediate code (e.g., Basic, Java, PASCAL, etc.) does not constitute code on the machine code level as in *Westheimer et al.* The significance of such difference between the present invention and what is taught in *Westheimer et al.* is better understood in view of the following.

The system of *Westheimer et al.* relies on the location of the operation code stored in the RAM 48 in order to determine or judge whether the code represents non-encoded operation code or encoded operation code. As is described in *Westheimer et al.* and unlike the presently claimed invention, whether a code stored in the RAM 48 is judged to be non-encoded code or encoded operation code is based on where the operation code is stored in the RAM 48. To wit, *Westheimer et al.* describes the TRANSFORM ENABLE signal

> *This arrangement causes the TRANSFORM ENABLE output of gate 36 on line 130 to be set high when an instruction fetch signal is generated by CPU 40 for an instruction stored within the address range defined by the upper and lower boundary addresses stored in the upper and lower boundary latches 24 and 26 [Fig. 1 reproduced above]... [T]he TRANSFORM ENABLE signal on line 130 can only be set high by an operation code which resides within the boundary address range defined by upper and lower boundary latches 24 and 26, i.e., by an operation code which is itself encoded... Data put out or read in by CPU 40 which falls*

outside said boundary address range will not undergo transformation. (See, e.g., Col. 6, ln. 37 to Col. 7, ln. 12).

Such operation of judging whether the operation code is encoded or not encoded in *Westheimer et al.* is illustrated in Figs. 4a and 4b (reproduced below), for example.

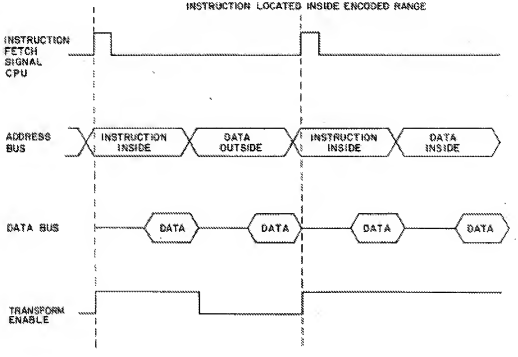

FIG. 4a

As is shown in Fig. 4a, if the CPU 40 fetches an instruction, it is judged or determined whether the instruction is located in the RAM 48 within a range representing the encoded operation code. If the operation code is judged to have an address inside the encoded range, the TRANSFORM ENABLE signal goes active to allow the fetched instruction to be decoded.

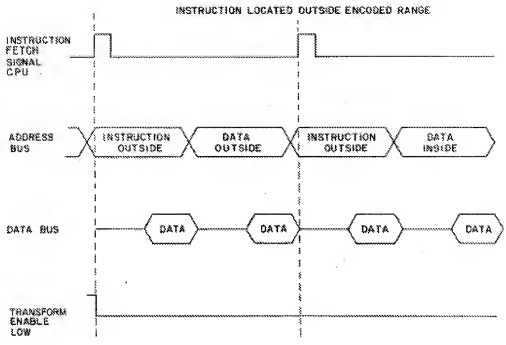

FIG. 4b

Fig. 4b illustrates the case where the fetched instruction is judged to have an address outside the encoded range. In such case the TRANSFORM ENABLE signal remains low or inactive. As a result, the fetched instruction does not undergo decoding.

Thus, the Examiner will undoubtedly appreciate that *Westheimer et al.* describes a system in which machine or operation code, *not intermediate code as in the present invention*, is either encoded or non-encoded within the RAM 48. Moreover, the operation code fetched from the RAM 48 *is judged to be encoded or non-encoded based on the location or address of the operation code within the RAM 48*. The present invention as defined in amended claim 1, on the other hand, judges whether the *intermediate code* is encrypted *independent of where the intermediate code is stored* in the RAM. Consequently, the present invention is much more suitable for allowing users to create and/or modify intermediate code within the LSI/optical disc with relative ease. The user need not be concerned with machine code instructions, where the instructions

may be stored in RAM, etc. Meanwhile, the encrypted intermediate code remains non-modifiable by a user.

Even more specifically, the present invention allows for relocation of the encrypted and non-encrypted intermediate code independent of location in RAM. The encrypted and non-encrypted intermediate code is not bound to a predefined address range in the manner of *Westheimer et al.* If the operation code in Westheimer et al. were to be simply relocated within the RAM, the system would no longer be operable in that it could no longer determine whether an operation code was encoded or not.

Buerkle et al. does not make up for the above-noted deficiencies. Namely, neither *Westheimer et al.* nor *Buerkle et al.*, taken alone or in combination, teach or suggest storing intermediate code and encrypted intermediate code in RAM in combination with judging whether the intermediate code is encrypted independent of where the intermediate code is stored in the RAM as recited in amended claim 1. Applicants respectfully request withdrawal of the rejection of claims 1 and 12.

II. REJECTION OF CLAIMS 4-14 UNDER 35 USC §103(a)

Claims 4-14 stand rejected under 35 USC §103(a) based on *Westheimer et al.* in view of *Buerkle et al.*, and further in view of *Hagiwara et al.* Applicants respectfully request withdrawal of this rejection for at least the following reasons.

Regarding independent claim 5, this claim has been amended to include features similar to amended claim 1 as discussed above. Accordingly, claim 5 may be distinguished over the teachings of *Westheimer et al.* and *Buerkle et al.* for at least the same reasons expressed above. Moreover, *Hagiwara et al.* does not make up for the above-discussed deficiencies in *Westheimer et al.* and *Buerkle et al.*

The remaining claims represent dependent claims that depend from either claim 1 or claim 5 discussed above, and may be distinguished for at least the same reasons.

III. INITIALED PTO-1449 FORMS

Applicants note that an Information Disclosure Statements (IDS) with accompanying PTO-1449 Form was filed on both October 8, 2004 and October 27, 2004. However, applicants have not been provided with initialed copies of the PTO-1449 forms indicating that the references cited therein have been considered. Instead, applicants received copies of the respective PTO-1449 Forms having a line drawn therethrough with the Official Action mailed on July 14, 2005. If such line was intended to indicate that the references have not been considered, applicant can find no indication in the record as to why such references may not have been considered.

Applicants respectfully request that the Examiner provide initialed copies of the PTO-1449 forms to ensure that the references appear on the cover of any resultant patent to indicate that the references have been considered.

Additionally, applicants note that the Examiner in the present Office Action now relies on USP 6,393,561 to Hagiwara et al. However, applicants are unable to find any indication in the record that Hagiwara et al. has officially been made of record (i.e, via a PTO-892 or PTO-1449 form). Applicants respectfully request that Hagiwara et al. be officially made of record to ensure that the reference appears on the cover of any resultant patent.

IV. CONCLUSION

Accordingly, all claims 1, 4-9 and 11-19 are believed to be allowable and the application is believed to be in condition for allowance. A prompt action to such end is earnestly solicited.

Should the Examiner feel that a telephone interview would be helpful to facilitate favorable prosecution of the above-identified application, the Examiner is invited to contact the undersigned at the telephone number provided below.

Should a petition for an extension of time be necessary for the timely reply to the outstanding Office Action (or if such a petition has been made and an additional extension is necessary), petition is hereby made and the Commissioner is authorized to charge any fees (including additional claim fees) to Deposit Account No. 18-0988.

Respectfully submitted,

RENNER, OTTO, BOISSELLE & SKLAR, LLP

/Mark D. Saralino/
Mark D. Saralino
Registration No. 34,243

DATE: February 16, 2007

The Keith Building
1621 Euclid Avenue
Nineteenth Floor
Cleveland, Ohio 44115
(216) 621-1113

R:\Mds\GEN\YAMA\Yamap797\yamap797amendmentnonfinal.wpd